**1.UNIT**

**What is a Database Management System?**

* ***Collection of data***
  + Interrelated data
  + Relevant to some endeavour
* ***Software to access the data***
  + Convenient
  + Efficient
* ***History***
  + 1950s-60s: magnetic tape and punched cards
  + 1960s-70s: hard disks, random access, file systems
  + 1970s-80s: relational model becoming competitive
  + 1980s-90s: relational model dominant, object-oriented databases
  + 1990s-00s: web databases and XML

**Why Study Databases?**

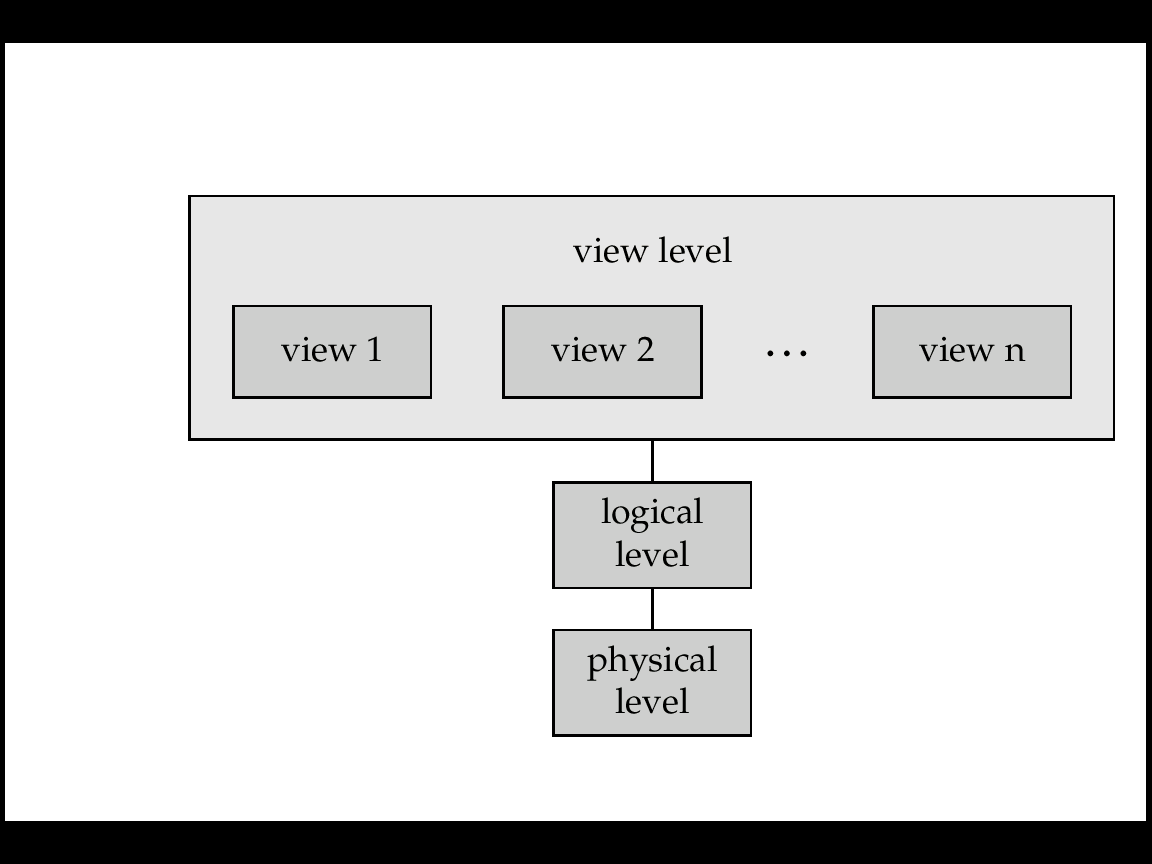
* They touch every aspect of our lives
* ***Applications:***
  + Banking: all transactions
  + Airlines: reservations, schedules
  + Universities: registration, course enrolment, grades
  + Sales: customers, products, purchases
  + Manufacturing: production, inventory, orders, supply chain
  + Human resources: employee records, salaries, tax deductions
  + Telecommunications: subscribers, usage, routing
  + Computer accounts: privileges, quotas, usage
  + Records: climate, stock market, library holdings
* ***Explosion of unstructured data on the web:***
  + Large document collections
  + Image databases, streaming media

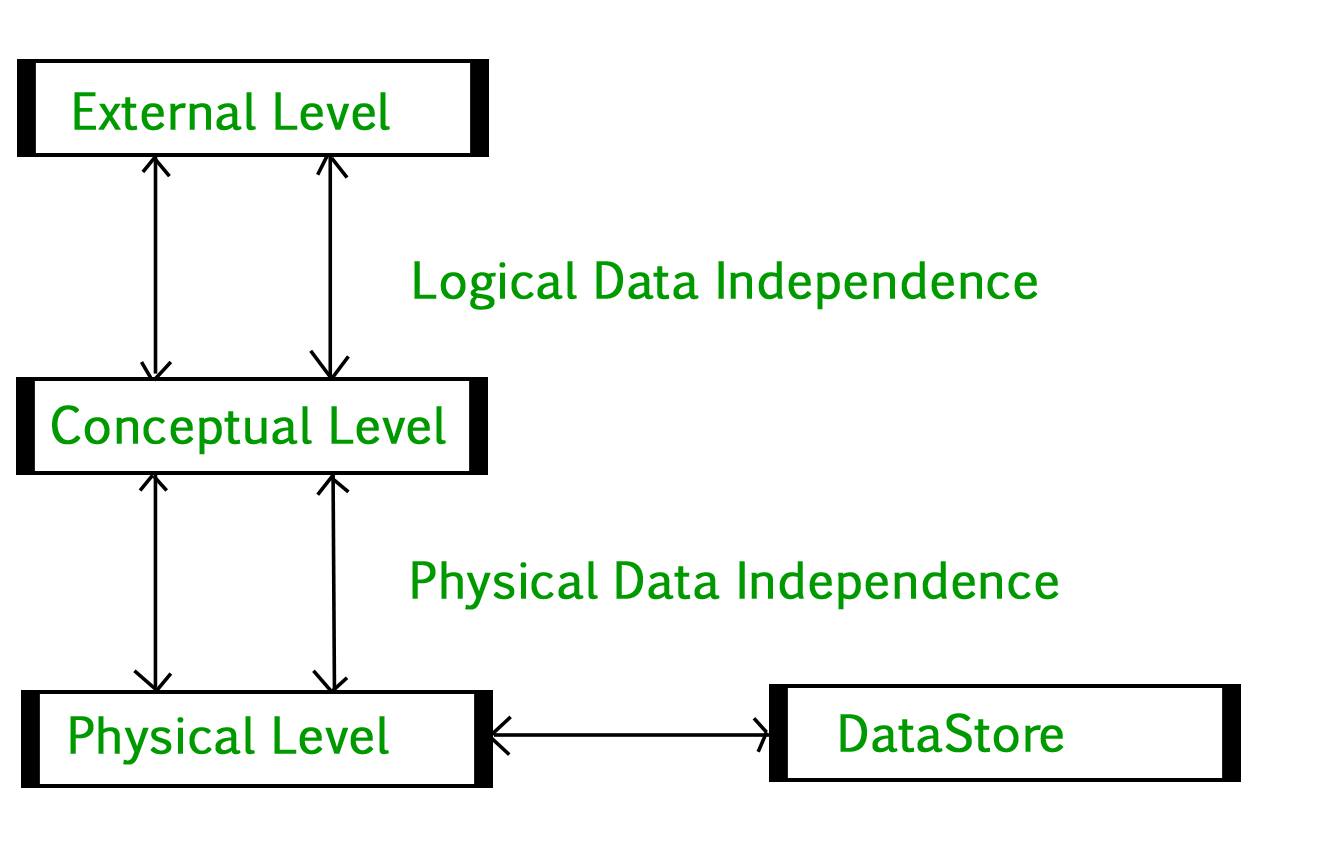
**1.2 Why not use file systems?**

* ***Data redundancy and inconsistency***
  + Multiple file formats
  + Duplication of information in different files
* ***Difficulty in accessing data*** 
  + Need to write a new program to carry out each new task
* ***Data isolation***
  + Multiple files and formats
* ***Integrity problems***
  + Integrity constraints (e.g. account balance > 0) become part of program code
  + Hard to add new constraints or change existing ones
* ***Maintenance problems***
  + When we add a new field, all existing applications must be modified to ignore it
* ***Atomicity of updates***
  + Failures may leave database in an inconsistent state with partial updates carried out
  + E.g. transfer of funds from one account to another should either complete or not happen at all
* ***Concurrent access by multiple users***
  + Concurrent accessed needed for performance
  + Uncontrolled concurrent accesses can lead to inconsistencies
    - E.g. two people reading a balance and updating it at the same time
  + Security problems
* ***Database systems offer solutions to all the above problems***

**The Levels of Abstraction**

* ***Physical level:*** how a record is stored on disk
* ***Logical level:*** describes data stored in database, and the relationships among the data.
* type customer = record  
   *name* : string;  
   *street* : string;  
   *city* : integer;  
   end;
* ***View level:*** application-specific selections and arrangements of the data
* hide details of data types
* Views can also hide information for security reasons



DBMS 3-tier architecture divides the complete system into three inter-related but independent modules as shown below: [](https://media.geeksforgeeks.org/wp-content/uploads/dbms-3tier.jpg)

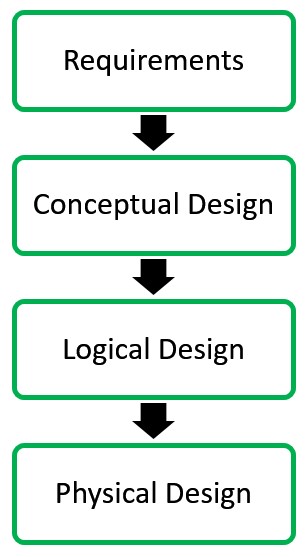
1. **Physical Level:** At the physical level, the information about the location of database objects in the data store is kept. Various users of DBMS are unaware of the locations of these objects.In simple terms,physical level of a database describes how the data is being stored in secondary storage devices like disks and tapes and also gives insights on additional storage details.
2. **Conceptual Level:**At conceptual level, data is represented in the form of various database tables. For Example, STUDENT database may contain STUDENT and COURSE tables which will be visible to users but users are unaware of their storage.Also referred as logical schema,it describes what kind of data is to be stored in the database.
3. **External Level:** An external level specifies a view of the data in terms of conceptual level tables.  Each external level view is used to cater to the needs of a particular category of users. For Example, FACULTY of a university is interested in looking course details of students, STUDENTS are interested in looking at all details related to academics, accounts, courses and hostel details as well. So, different views can be generated for different users. The main focus of external level is data abstraction.

**Data Independence**

Data independence means a change of data at one level should not affect another level. Two types of data independence are present in this architecture:

1. **Physical Data Independence:** Any change in the physical location of tables and indexes should not affect the conceptual level or external view of data. This data independence is easy to achieve and implemented by most of the DBMS.
2. **Conceptual Data Independence:** The data at conceptual level schema and external level schema must be independent. This means a change in conceptual schema should not affect external schema. e.g.; Adding or deleting attributes of a table should not affect the user’s view of the table. But this type of independence is difficult to achieve as compared to physical data independence because the changes in conceptual schema are reflected in the user’s view.

**Phases of database design**

Database designing for a real-world application starts from capturing the requirements to physical implementation using DBMS software which consists of following steps shown below: [](https://media.geeksforgeeks.org/wp-content/uploads/20190506212511/dbms-phases.jpg) **Conceptual Design:**The requirements of database are captured using high level conceptual data model. For Example, the ER model is used for the conceptual design of the database. **Logical Design:**Logical Design represents data in the form of relational model. ER diagram produced in the conceptual design phase is used to convert the data into the Relational Model. **Physical Design:** In physical design, data in relational model is implemented using commercial DBMS like Oracle, DB2.

**Advantages of DBMS**

DBMS helps in efficient organization of data in database which has following advantages over typical file system:

* **Minimized redundancy and data inconsistency:** Data is normalized in DBMS to minimize the redundancy which helps in keeping data consistent. For Example, student information can be kept at one place in DBMS and accessed by different users.This minimized redundancy is due to primary key and foreign keys
* **Simplified Data Access:** A user need only name of the relation not exact location to access data, so the process is very simple.
* **Multiple data views:** Different views of same data can be created to cater the needs of different users. For Example, faculty salary information can be hidden from student view of data but shown in admin view.
* **Data Security:** Only authorized users are allowed to access the data in DBMS. Also, data can be encrypted by DBMS which makes it secure.
* **Concurrent access to data:**Data can be accessed concurrently by different users at same time in DBMS.
* **Backup and Recovery mechanism:** DBMS backup and recovery mechanism helps to avoid data loss and data inconsistency in case of catastrophic failures.

Also see

* [All DBMS Articles](https://www.geeksforgeeks.org/category/dbms/)
* [DBMS Quizzes](https://www.geeksforgeeks.org/quiz-corner-gq/)

In DBMS, the 3-tier architecture is a client-server architecture that separates the user interface, application processing, and data management into three distinct tiers or layers. The 3-tier architecture is widely used in modern web applications and enterprise systems because it offers scalability, flexibility, and security. Here is a brief description of each tier in the 3-tier architecture:

1. Presentation Tier: The presentation tier is the user interface or client layer of the application. It is responsible for presenting data to the user and receiving input from the user. This tier can be a web browser, mobile app, or desktop application.
2. Application Tier: The application tier is the middle layer of the 3-tier architecture. It is responsible for processing and managing the business logic of the application. This tier communicates with the presentation tier to receive user input and communicates with the data management tier to retrieve or store data. This tier may include application servers, web servers, or APIs.
3. Data Management Tier: The data management tier is the bottom layer of the 3-tier architecture. It is responsible for managing and storing data. This tier can include databases, data warehouses, or data lakes. The data management tier communicates with the application tier to receive or store data.
4. The 3-tier architecture in DBMS provides several benefits, including:
5. Scalability: The architecture separates the application processing and data management layers, which allows for easy scalability of each layer independently.
6. Flexibility: The architecture allows for the replacement or upgrade of one layer without affecting the other layers.
7. Security: The architecture provides an additional layer of security, as the data management tier can be isolated from the application and presentation tiers, reducing the risk of unauthorized access.

Overall, the 3-tier architecture in DBMS is a flexible, scalable, and secure approach to building modern web applications and enterprise systems. It separates the user interface, application processing, and data management into distinct layers, providing clear boundaries between each layer and improving system performance, reliability, and maintainability.

**Database Administrator:**

**A Database Administrator** (DBA) is an individual or person responsible for controlling, maintaining, coordinating, and operating a database management system. Managing, securing, and taking care of the database systems is a prime responsibility. They are responsible and in charge of authorizing access to the database, coordinating, capacity, planning, installation, and monitoring uses, and acquiring and gathering software and hardware resources as and when needed. Their role also varies from configuration, database design, migration, security, troubleshooting, backup, and data recovery. Database administration is a major and key function in any firm or organization that is relying on one or more databases. They are overall commanders of the Database system.

Types of Database Administrator (DBA) :

Administrative DBA –

Their job is to maintain the server and keep it functional. They are concerned with data backups, security, troubleshooting, replication, migration, etc.

Data Warehouse DBA –

Assigned earlier roles, but held accountable for merging data from various sources into the data warehouse. They also design the warehouse, with cleaning and scrubs data prior to loading.

Cloud DBA –

Nowadays companies are preferring to save their workpiece on cloud storage. As it reduces the chance of data loss and provides an extra layer of data security and integrity.

Development DBA –

They build and develop queries, stores procedure, etc. that meets firm or organization needs. They are par at programming.

Application DBA –

They particularly manage all requirements of application components that interact with the database and accomplish activities such as application installation and coordination, application upgrades, database cloning, data load process management, etc.

Architect –

They are held responsible for designing schemas like building tables. They work to build a structure that meets organizational needs. The design is further used by developers and development DBAs to design and implement real applications.

OLAP DBA –

They design and build multi-dimensional cubes for determination support or OLAP systems.

Data Modeler –

In general, a data modeler is in charge of a portion of a data architect’s duties. A data modeler is typically not regarded as a DBA, but this is not a hard and fast rule.

Task-Oriented DBA –

To concentrate on a specific DBA task, large businesses may hire highly specialised DBAs. They are quite uncommon outside of big corporations. Recovery and backup DBA, whose responsibility it is to guarantee that the databases of businesses can be recovered, is an example of a task-oriented DBA. However, this specialism is not present in the majority of firms. These task-oriented DBAs will make sure that highly qualified professionals are working on crucial DBA tasks when it is possible.

Database Analyst –

This position doesn’t actually have a set definition. Junior DBAs may occasionally be referred to as database analysts. A database analyst occasionally performs functions that are comparable to those of a database architect. The term “Data Administrator” is also used to describe database analysts and data analysts. Additionally, some businesses occasionally refer to database administrators as data analysts.

Importance of Database Administrator (DBA) :

Database Administrator manages and controls three levels of database internal level, conceptual level, and external level of Database management system architecture and in discussion with the comprehensive user community, gives a definition of the world view of the database. It then provides an external view of different users and applications.

Database Administrator ensures held responsible to maintain integrity and security of database restricting from unauthorized users. It grants permission to users of the database and contains a profile of each and every user in the database.

Database Administrators are also held accountable that the database is protected and secured and that any chance of data loss keeps at a minimum.

Database Administrator is solely responsible for reducing the risk of data loss as it backup the data at regular intervals.

Role and Duties of Database Administrator (DBA) :

Decides hardware –

They decide on economical hardware, based on cost, performance, and efficiency of hardware, and best suits the organization. It is hardware that is an interface between end users and the database.

Manages data integrity and security –

Data integrity needs to be checked and managed accurately as it protects and restricts data from unauthorized use. DBA eyes on relationships within data to maintain data integrity.

Database Accessibility –

Database Administrator is solely responsible for giving permission to access data available in the database. It also makes sure who has the right to change the content.

Database design –

DBA is held responsible and accountable for logical, physical design, external model design, and integrity and security control.

Database implementation –

DBA implements DBMS and checks database loading at the time of its implementation.

Query processing performance –

DBA enhances query processing by improving speed, performance, and accuracy.

Tuning Database Performance –

If the user is not able to get data speedily and accurately then it may lose organization’s business. So by tuning SQL commands DBA can enhance the performance of the database.

Various responsibilities of Database Administrator (DBA) :

Responsible for designing overall database schema (tables & fields).

To select and install database software and hardware.

Responsible for deciding on access methods and data storage.

DBA selects appropriate DBMS software like oracle, SQL server or MySQL.

Used in designing recovery procedures.

DBA decides the user access level and security checks for accessing, modifying or manipulating data.

DBA is responsible for specifying various techniques for monitoring the database performance.

DBA is responsible for operation managements.

The operation management deals with the data problems which arises on day to day basis, and the responsibilities include are:

Investigating if any error is been found in the data.

Supervising of restart and recovery procedures in case of any event failure.

Supervising reorganization of the databases.

Controlling and handling all periodic dumps of data.

Skills Required for DBA:

1. The various programming and soft skills are required to DBA are as follows,

Good communication skills

Excellent knowledge of databases architecture and design and RDBMS.

Knowledge of Structured Query Language (SQL).

2. In addition, this aspect of database administration includes maintenance of data security, which involves maintaining security authorization tables, conducting periodic security audits, investigating all known security breaches.

3. To carry out all these functions, it is crucial that the DBA has all the accurate information about the company’s data readily on hand. For this purpose he maintains a data dictionary.

4. The data dictionary contains definitions of all data items and structures, the various schemes, the relevant authorization and validation checks and the different mapping definitions.

5. It should also have information about the source and destination of a data item and the flow of a data item as it is used by a system. This type of information is a great help to the DBA in maintaining centralized control of data.

**CLIENT-SERVER ARCHITECTURE**:

A Database store a lot of critical information to access data quickly and securely. Hence it is important to select the correct architecture for efficient data management. DBMS Architecture helps users to get their requests done while connecting to the database. We choose database architecture depending on several factors like the size of the database, number of users, and relationships between the users. There are two types of database models that we generally use, are logical model and physical model. Several types of architecture are there in the database which we will deal with in the next section.

**Types of DBMS Architecture**

There are several types of DBMS Architecture that we use according to the usage requirements. Types of DBMS Architecture are discussed here.

* 1-Tier Architecture
* 2-Tier Architecture
* [3-Tier Architecture](https://www.geeksforgeeks.org/introduction-of-3-tier-architecture-in-dbms-set-2/)

**1-Tier Architecture**

In 1-Tier Architecture the database is directly available to the user, the user can directly sit on the DBMS and use it that is, the client, server, and Database are all present on the same machine. For Example: to learn SQL we set up an SQL server and the database on the local system. This enables us to directly interact with the relational database and execute operations. The industry won’t use this architecture they logically go for 2-Tier and 3-Tier Architecture.

*DBMS 1-Tier Architecture*

**Advantages of 1-Tier Architecture**

Below mentioned are the advantages of 1-Tier Architecture.

* **Simple Architecture:** 1-Tier Architecture is the most simple architecture to set up, as only a single machine is required to maintain it.
* **Cost-Effective:** No additional hardware is required for implementing 1-Tier Architecture, which makes it cost-effective.
* **Easy to Implement:**1-Tier Architecture can be easily deployed, and hence it is mostly used in small projects.

**2-Tier Architecture**

The 2-tier architecture is similar to a basic[client-server model](https://www.geeksforgeeks.org/client-server-model/). The application at the client end directly communicates with the database on the server side. APIs like ODBC and JDBC are used for this interaction. The server side is responsible for providing query processing and transaction management functionalities. On the client side, the user interfaces and application programs are run. The application on the client side establishes a connection with the server side in order to communicate with the DBMS.   
An advantage of this type is that maintenance and understanding are easier, and compatible with existing systems. However, this model gives poor performance when there are a large number of users.



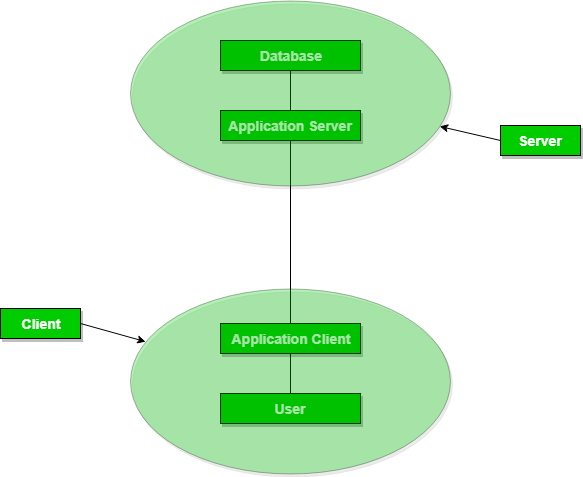
*DBMS 2-Tier Architecture*

**Advantages of 2-Tier Architecture**

* **Easy to Access:** 2-Tier Architecture makes easy access to the database, which makes fast retrieval.
* **Scalable:** We can scale the database easily, by adding clients or by upgrading hardware.
* **Low Cost:** 2-Tier Architecture is cheaper than 3-Tier Architecture and [Multi-Tier Architecture](https://www.geeksforgeeks.org/multi-tier-architecture-of-data-warehouse/).
* **Easy Deployment:** 2-Tier Architecture is easy to deploy than 3-Tier Architecture.
* **Simple:** 2-Tier Architecture is easily understandable as well as simple because of only two components.

**3-Tier Architecture**

In [3-Tier Architecture](https://www.geeksforgeeks.org/introduction-of-3-tier-architecture-in-dbms-set-2/), there is another layer between the client and the server. The client does not directly communicate with the server. Instead, it interacts with an application server which further communicates with the database system and then the query processing and transaction management takes place. This intermediate layer acts as a medium for the exchange of partially processed data between the server and the client. This type of architecture is used in the case of large web applications.



*DBMS 3-Tier Architecture*

**Advantages of 3-Tier Architecture**

* **Enhanced scalability:** Scalability is enhanced due to distributed deployment of application servers. Now, individual connections need not be made between the client and server.
* **Data Integrity:** 3-Tier Architecture maintains Data Integrity. Since there is a middle layer between the client and the server, data corruption can be avoided/removed.
* **Security:**3-Tier Architecture Improves Security. This type of model prevents direct interaction of the client with the server thereby reducing access to unauthorized data.

**Disadvantages of 3-Tier Architecture**

* **More Complex:**3-Tier Architecture is more complex in comparison to 2-Tier Architecture. Communication Points are also doubled in 3-Tier Architecture.
* **Difficult to Interact:** It becomes difficult for this sort of interaction to take place due to the presence of middle layers.